SAND2015-6813PE

Dakota Software Training

Model Calibration

http://dakota.sandia.gov

Exceptional service in the national interest

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Module Learning Goals

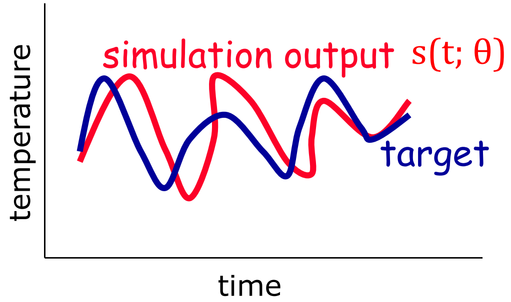
In this module you will learn

- Why you might want to tune models to match data via calibration (parameter estimation)
- How to formulate calibration problems and present them to Dakota
- What Dakota methods can help you achieve calibration goals

Exercise: create a Dakota calibration study and try to infer unknown parameters for a synthetic data set.

Calibration: Fitting Models to Data

- Use data to improve characterization of input parameter values, by maximizing agreement between simulation output and experiment target
  - Infer unknown conditions, source terms, or properties
  - Tune models to match specific scenarios
  - Make them more robust to predict a range of outcomes



- Also known as parameter estimation/identification, inverse modeling
- Can also calibrate one model to another (typically higher fidelity) model

➢ Calibration is not validation! Separate hold-out data must be used to assess whether a calibrated model is valid.

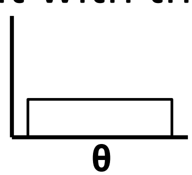
Classes of Model Calibration

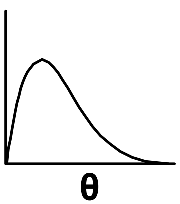
- Goal: maximize agreement between observations y_i and corresponding simulation output $s_i(\theta)$; typically a nonlinear, implicit function of θ (parameterized simulation)

- Deterministic calibration: seek one or more sets of parameter values that best match the data y, typically in the two-norm:

$$\min_{\theta} f(\theta) = SSE(\boldsymbol{\theta}) = \sum_{i=1}^{n} [(s_i(\boldsymbol{\theta}) - y_i]^2 = \sum_{i=1}^{n} [r_i(\boldsymbol{\theta})]^2$$

  - Least-squares: initial iterate θ_0, *nonlinear optimization*, updated values θ
- Statistical calibration: seek a statistical characterization of parameters most consistent with the data

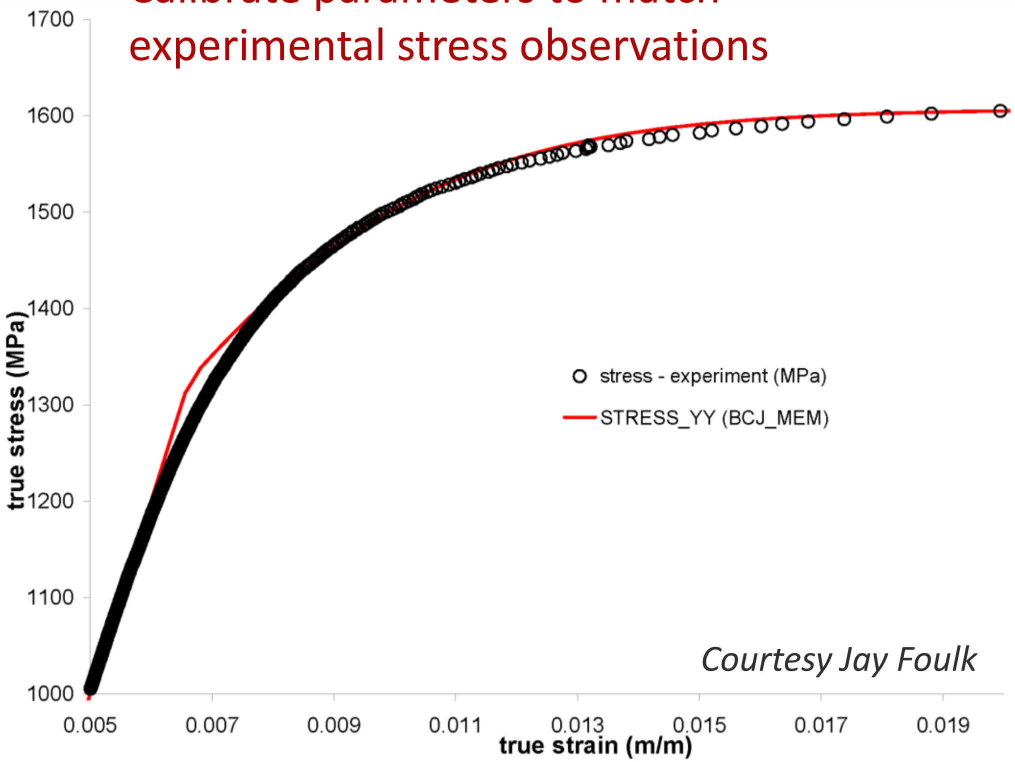




  - Bayesian: prior distribution, *statistical inference (MCMC)*, posterior distribution

Example: Parameter Estimation for a Material Plasticity Model

Calibrate parameters to match experimental stress observations



Courtesy Jay Foulk

f – yields rate dependence (fit)
Y – the yield stress (chosen)
n – exponent in flow rule (fit)
H – hardening in evolution of κ (fit)
R_d – recovery in evolution of κ (fit)

f	*4.52 x 10^4*
Y	*1325 MPa*
n	*0.386*
H	*1.10 x 10^5 MPa*
R_d	*389*

NOTE: Experimental data taken from a representative test, ph13-8-h950-test-3

Flow rule concentrating the effective stress

$$\dot{\epsilon_p} = f\{\sinh[\frac{\bar{\sigma}}{(1-\phi)(\kappa+Y)} - 1]\}^n$$

evolution of isotropic hardening

$$\dot{\kappa} = [H - R_d\kappa]\dot{\epsilon_p}$$

*Large values of f make the formulation rate independent. I did not need to fit f.

Specifying Calibration Parameters

- Deterministic calibration problems are presented to Dakota using design variables (same as optimization)
- Initial point starts the solve for local methods
- Bounds for the search are typical, but not required for all methods
- See advanced slides for Bayesian methods, which use uncertain variables instead of design

Cantilever calibration variable example

```
variables

  # calibration parameters
  continuous_design 3
    upper_bounds  3.1e7 10   10
    initial_point 2.9e7 4    4
    lower_bounds  2.7e7 1    1
    descriptors   'E'   'w'  't'

  # Fixed config parameters
  continuous_state 3
    initial_state 40000 500 1000
    descriptors 'R' 'X' 'Y'
```

Defining Calibration Responses

$$\min_{\theta} f(\theta) = SSE(\boldsymbol{\theta}) = \sum_{i=1}^{n} [(s_i(\boldsymbol{\theta}) - y_i]^2 = \sum_{i=1}^{n} [r_i(\boldsymbol{\theta})]^2$$

Three main options:

1. Interface returns differences (residuals) $r_i(\theta) = s_i(\theta) - y_i$ to Dakota

```
responses
  calibration_terms = 2
  descriptors
    'stress_diff' 'displ_diff'
```

2. Interface returns simulation outputs $s_i(\theta)$ to Dakota; specify data file containing y_i values

```
responses
  calibration_terms = 2
  descriptors
    'sim_stress' 'sim_displ'
  calibration_data_file  'myobs.dat'
    num_experiments = 3
```

3. Interface returns composite objective $f(\theta)$; gives advanced users greater control

```
responses
  objective_functions = 1
  descriptors  'f_SSE'
```

Local nonlinear least squares methods require set of residuals (Option 1 or 2)

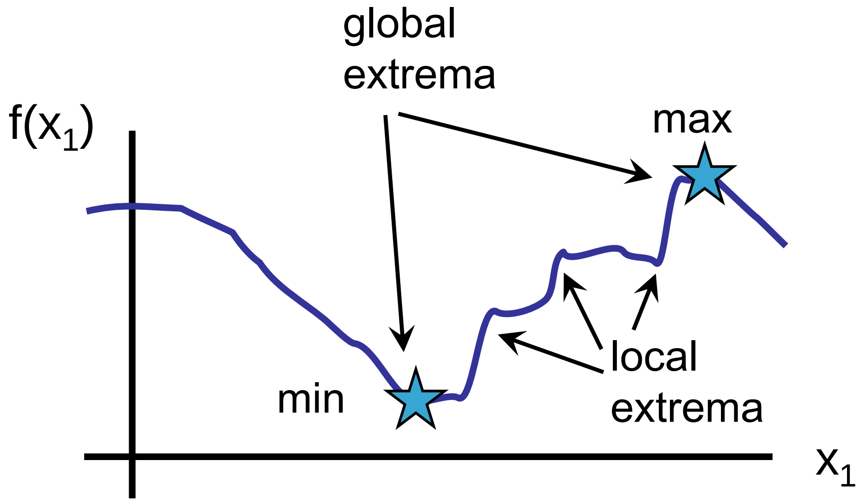
Dakota Calibration Methods

Deterministic

- For local parameter value improvement; reliable simulation derivatives: specialized local least-squares solvers
- Local search with unreliable derivatives: pattern search
- Global best parameter set: global optimizers such as DiRECT or genetic algorithms (can be costly)
- Other advanced optimization approaches

Statistical

- Calibrate distribution parameters to match data: any of the above solvers with a nested model
- Bayesian inference: Markov Chain Monte Carlo (QUESO)



Classes of Methods

Gradient Descent

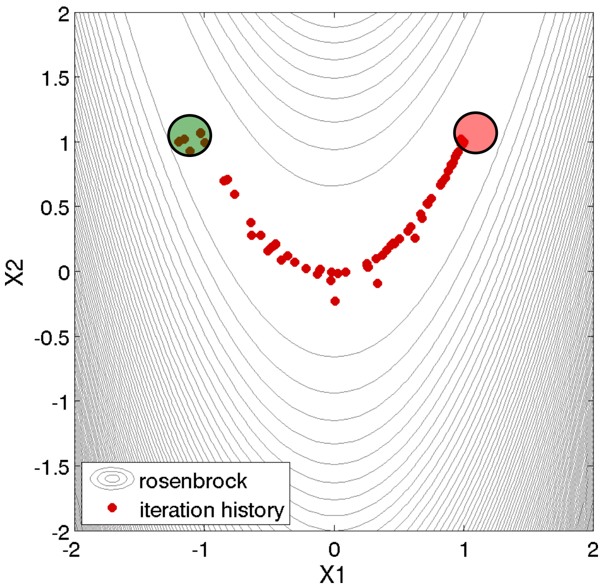
- Looks for improvement based on derivative
- Requires analytic or numerical derivatives
- Efficient/scalable for smooth problems
- Converges to local extreme

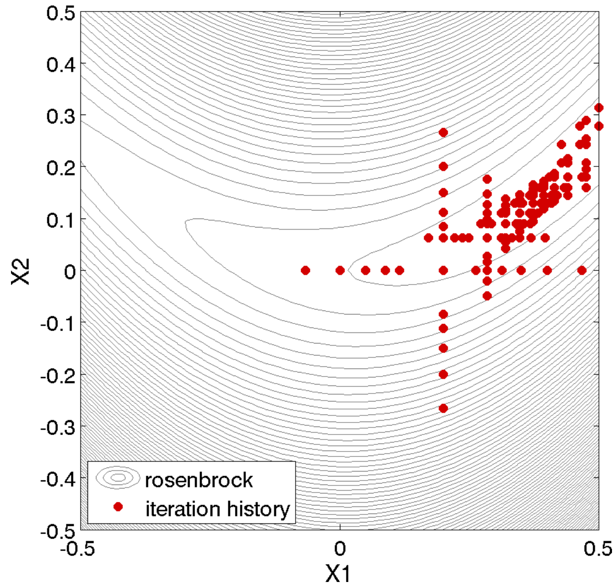
Derivative-Free Local

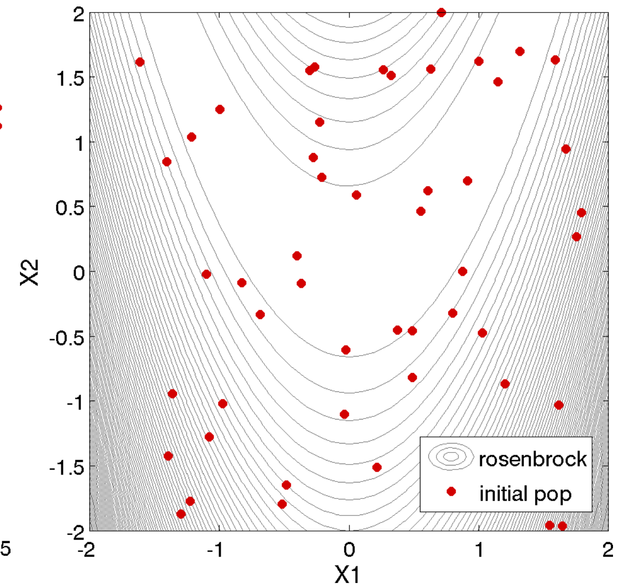
- Sampling with bias/rules toward improvement
- Requires only function values
- Good for noisy, unreliable or expensive derivatives
- Converges to local extreme

Derivative-Free Global

- Broad exploration with selective exploitation
- Requires only function values
- Typically computationally intensive
- Converges to global extreme







More About Local Calibration

- Local, derivative-based least squares solvers are similar to Newton methods for general nonlinear programming
- They can take advantage of the squared residual formulation

$$\frac{SSE}{2} = f(\theta) = \frac{1}{2} r(\theta)^T r(\theta) = \frac{1}{2} [s(\theta) - y]^T [s(\theta) - y]$$

$$\nabla f(\theta) = J(\theta)^T r(\theta); \quad J_{ij} = \frac{\partial r_i}{\partial \theta_j} \quad \nabla^2 f(\theta) = J^T J + \sum_{i=1}^{n} r_i(\theta) \nabla^2 r_i(\theta)$$

  and either ignore the circled Hessian term (as residuals should be small as the algorithm converges), or successively approximate it during optimization
- Dakota's NL2SOL local calibration algorithm uses a quasi-Newton update scheme to approximate the Hessian, and is often more robust than other solvers when the residuals are not small.
- These methods can be very efficient, converging in a few function evaluations

Exercise: Find Beam Properties

- The directory `~/exercises/calibration` contains data files with observations of mass, displacement, and stress from beam experiments
- As experiments were conducted, the observation error was reduced by improving the measurement equipment. File extensions .1--.5 correspond to 0.5, 0.1, 0.05, 0.01, and 0.0 relative error.
- Complete the Dakota input file `dakota_calibration_sketch.in` to use NL2SOL to determine the properties (Young's modulus E, width w, and thickness t) of the beam used in the experiment. Hold R, X, Y fixed.

Hints:

- Previous example input files can help with the variables blocks
- See the reference manual sections on:
  - Variables: continuous design, continuous state
  - Responses: calibration_terms (the simulator returns the predicted QOIs), calibration data file and its format, gradient types
  - Scaling (method, variables, responses)

Exercise: Find Beam Properties

- How do your estimated parameter values compare to your neighbors?
- Is it sensitive to the initial point?
- Do your parameter estimates converge as the noise level in the data is reduced (data files .1 through .5)?
- What do you observe in the final residuals, SSE, and confidence intervals?
- What happens if you use a pattern search or DiRECT method?

Parameter Identifiability

- Looking at the cantilever beam equations, which parameters would you expect to be able to estimate given data on which responses?
- How would you determine this for an implicit function (black-box simulation)?

$$M = \rho * wt * \frac{L}{12^3}$$

$$S = \frac{600}{wt^2} Y + \frac{600}{w^2 t} X$$

$$D = \frac{4L^3}{Ewt} \sqrt{\left(\frac{Y}{t^2}\right)^2 + \left(\frac{X}{w^2}\right)^2}$$

Guide to Calibration Methods

Category	Dakota method names	Continuous Variables	Categorical/ Discrete Variables	Bound Constraints	General Constraints
Gradient-Based Local (Smooth Response)	`nl2sol`	x		x	
	`nlssol_sqp, optpp_g_newton`	x		x	x
Gradient-Based Global (Smooth Response)	`hybrid strategy, multi_start strategy`	x		x	x
Derivative-Free Global (Nonsmooth Response)	`efficient_global, surrogate_based_global`	x		x	x

See Usage Guidelines in Dakota User's Manual.

Also, can apply any optimizer when doing derivative-free local or global calibration.

Calibration References

- G. A. F. Seber and C. J. Wilde, “Nonlinear Regression”, John Wiley and Sons, Inc., Hoboken, New Jersey, 2003.

- M. C. Hill and C. R. Tiedeman, “Effective Groundwater Model Calibration: With Analysis of Data, Sensitivities, Predictions, and Uncertainty”, John Wiley and Sons, Inc., Hoboken, New Jersey, 2007.

- R. C. Aster, B. Borchers, and C. H. Thurber, “Parameter Estimation and Inverse Problems”, Elsevier, Inc., Oxford, UK, 2005.

- Dakota User’s Manual
  - Nonlinear Least Squares Capabilities
  - Surrogate-Based Minimization
- Dakota Reference Manual

Guide to Optimization Methods

Category	Dakota method names	Continuous Variables	Categorical/ Discrete Variables	Bound Constraints	General Constraints
Gradient-Based Local (Smooth Response)	optpp_cg	x			
	dot_bfgs, dot_frcg, conmin_frcg	x		x	
	npsol_sqp, nlpql_sqp, dot_mmfd, dot_slp, dot_sqp, conmin_mfd, optpp_newton, optpp_q_newton, optpp_fd_newton, weighted sums (multiobjective), pareto_set strategy (multiobjective)	x		x	x
Gradient-Based Global (Smooth Response)	hybrid strategy, multi_start strategy	x		x	x
Derivative-Free Local (Nonsmooth Response)	optpp_pds	x		x	
	asynch_pattern_search, coliny_cobyla, coliny_pattern_search, coliny_solis_wets, surrogate_based_local	x		x	x
Derivative-Free Global (Nonsmooth Response)	ncsu_direct	x		x	
	coliny_direct, efficient_global, surrogate_based_global	x		x	x
	coliny_ea, soga, moga (multiobjective)	x	x	x	x

See Usage Guidelines in Dakota User's Manual

Optimization References

- J. Nocedal and S. J. Wright, “Numerical Optimization”, Second Edition, Springer Science and Business Media, LLC, New York, New York, 2006.

- S. S. Rao, “Engineering Optimization: Theory and Practice”, Fourth Edition, John Wiley and Sons, Inc., Hoboken, New Jersey, 2009.

- Dakota User’s Manual
  - Optimization Capabilities
  - Surrogate-Based Minimization
  - Advanced Strategies
  - Advanced Model Recursions: Optimization Under Uncertainty
- Dakota Reference Manual

APPLICATION-SPECIFIC EXAMPLE